Machine learning applications in proteomics research: how the past can boost the future.

Journal:	*PROTEOMICS*
Manuscript ID:	pmic.201300289.R1
Wiley - Manuscript type:	Review
Date Submitted by the Author:	n/a

Complete List of Authors:	Kelchtermans, Pieter; VIB, Dept. Medical Protein Research; Ghent University, Dept. of Biochemistry Bittremieux, Wout; University of Antwerp, Department of Mathematics and Computer Science; University of Antwerp, Biomedical Informatics Research Center Antwerp (biomina) De Grave, Kurt; KU Leuven, Department of Computer Science Degroeve, Sven; Universiteit Gent, Dept. Biochemistry; VIB, Dept. Med Prot Res Ramon, Jan; KU Leuven, Department of Computer Science Laukens, Kris; University of Antwerp, Department of Mathematics and Computer Science; University of Antwerp, Biomedical Informatics Research Center Antwerp (biomina) Valkenborg, Dirk; Flemish Institute for Technological Research (VITO), ; University of Antwerp, CFP-CeProMa Barsnes, Harald; University of Bergen, Proteomics Unit, Department of Biomedicine Martens, Lennart; Universiteit Gent, Department of Biochemistry; VIB, Department of Medical Protein Research

Keywords (free text entry):	

Machine learning applications in proteomics research: how the past can boost the future.

Pieter Kelchtermans[1,2,7], Wout Bittremieux[3,4], Kurt De Grave[5], Sven Degroeve[1,2],Jan Ramon[5], Kris Laukens[3,4],Dirk Valkenborg[6,7,8], Harald Barsnes[9], Lennart Martens[1,2].

[1] Department of Medical Protein Research, VIB, B-9000 Ghent, Belgium
[2] Department of Biochemistry, Faculty of Medicine and Health Sciences, Ghent University, Ghent, Belgium
[3] Department of Mathematics and Computer Science, University of Antwerp, Antwerp, Belgium
[4] Biomedical Informatics Research Center Antwerp (biomina), University of Antwerp, Antwerp, Belgium
[5] Department of Computer Science, KU Leuven, Celestijnenlaan 200A, B-3001 Heverlee, Belgium
[6] I-BioStat, Hasselt University, Belgium
[7] Flemish Institute for Technological Research (VITO), Boeretang 200, B-2400 Mol Belgium
[8] CFP-CeProMa, University of Antwerp, Belgium
[9] Proteomics Unit, Department of Biomedicine, University of Bergen, Norway

Corresponding author:

Prof. Dr. Lennart Martens, Department of Medical Protein Research and Biochemistry, VIB and Faculty of Medicine and Health Sciences, Ghent University, A. Baertsoenkaai 3, B-9000 Ghent, Belgium. Email: lennart.martens@ugent.be. Tel: 32-92649458 Fax: 32-92649484.

Abbreviations: ROC: receiver operating characteristic; SVM: support vector machine. PSM: peptide-spectrum match

Keywords: machine learning, pattern recognition, shotgun proteomics, standardization.

Total number of words: ...

Abstract

Machine learning is a subdiscipline within artificial intelligence that focuses on algorithms that allow computers to learn solving a (complex) problem from existing data. This ability can be used to generate a solution to a particularly intractable problem, given that enough data is available to train and subsequently evaluate an algorithm on. Since mass spectrometry based proteomics has no shortage of complex problems, and since publicly available data is becoming available in ever growing amounts, machine learning is fast becoming a very popular tool in the field. We here therefore present an overview of the different applications of machine learning in proteomics that together cover nearly the entire wet- and dry-lab workflow, and that address key bottlenecks in experiment planning and design, as well as in data processing and analysis.

Main Text

1.0 Proteomics and machine learning

1.1 Background

This review focuses on the application of predictive models in mass spectrometry based proteomics, structured by the specific applications in the field. Machine learning models have also been employed elsewhere in the life sciences, for instance in the related context of protein-protein interactions as reviewed by Hooda *et al*. [1] and by Srihari *et al*. [2], and for the analysis of cellular systems or biomarker discovery as reviewed by Elo and Schwikowski [3]. This review aims to complement and extend these existing efforts by providing an in-depth study of the specific tools used in mass spectrometry based proteomics. First we introduce the field of machine learning and highlight the essential concepts. Then specific implementations of machine learning algorithms are reviewed, ordered by the steps in a typical proteomics workflow these implementations cover.

Earlier efforts focused primarily on algorithms to classify or differentiate between two groups of subjects based on high-throughput profiling analyses [4] [5] [6] [7].The field has since moved to more specific algorithms that target individual steps in the nowadays much more commonly used shotgun or targeted proteomics workflows [8] [9] [10] [11]. It is these latter type of applications that will form the core focus of this review.

1.2 Machine Learning

Machine learning is the subfield of artificial intelligence that studies how computers can learn. It relies heavily on techniques and theory from statistics, optimization, algorithms, and in some cases biology. A machine learns—improves its performance to solve a certain problem—by receiving information about the problem at hand, often observations of how example cases of

the problem were solved in the past. In order to learn, machine learning algorithms find exploitable regularities. Machine learning is therefore closely related to pattern recognition [12] and data mining. Several good introductions to the field are available [13] [14] [15] .

The archetypal machine learning setting is *supervised learning*. In this setting, we assume an unknown but fixed function $y = f(x)$ or in non-deterministic cases an unknown probability distribution $\mathrm{P(y|x)}$ between the input *features* x (the information available about a certain case of the problem) and the corresponding output *label* y (the solution of that case of the problem). The task of the learner is to learn this relationship, receiving a number of correct training examples cases (instances) $(x_i, y_i)$with $y_i = f(x_i)$as input, to predict (or explain) the label of unseen, unlabeled examples x^* with a minimal expected loss. The loss can be the fraction of incorrect predictions, the mean square error, or some similar measure.

For example, suppose we want to predict where the protease trypsin will cleave a protein during digestion. We can make a list of potential cleavage positions (cases), and assign a label to each position in the protein sequence for which we have identified peptides in an LC/MS experiment, saying whether it results from a cleavage or non-cleavage at that position. In other words, we construct example (x, y)-pairs with x a set of features about the cleavage site in the protein and y a label “cleaved” or “uncleaved”. A learning algorithm can investigate features of the cleavage sites such as “Is there a tryptophan residue two positions N-terminal of the cleavage position?” and find correlations between those features and the success of the protease [16]. Out of that information it then builds a model for predicting the probability of successful cleavage for potential cleavage sites in novel proteins.

Supervised learning tasks can be classified based on the type of the labels. When the labels are discrete, e.g., $\mathrm{y} \in \{cleaved, uncleaved\}$, the learning task is called *classification*. When the predicted labels are continuous however, such as concentrations or spectral peak intensities, the task is called *regression*.

S*emi-supervised learning* methods learn from a collection of cases where only part of them have been assigned labels. This approach has seen a surge of interest recently since one only needs to manually or experimentally label a limited set of examples, and unlabeled instances can still help the learning algorithm to better model the distribution. The algorithm can exploit this additional, unlabeled information by making the assumption that the class boundary tends not to lie in densely populated areas of the feature set (which holds in many applications). **Figure 1** illustrates this idea. A well-known example of a semi-supervised learning algorithm from the field of proteomics is Percolator [17], which classifies peptide-to-spectrum matches (PSMs) iteratively based on a set in which only the best (in the first iteration these are PSMs that match above a certain preset search engine score confidence) and the worst (matches from a decoy or nonsense database) cases are labeled.

Finally, *unsupervised learning* is a type of learning where examples are not labeled. Typical tasks in this setting include clustering (often used in exploratory research where a large amount of data has to be structured) and anomaly detection (amongst others useful for quality control).

Orthogonally, learning tasks can also be classified depending on the amount of structure present in the examples. The most basic setting is *attribute-value learning*, where an example can be represented as a row in a table. Three commonly used paradigms exist that allow a richer representation of examples: relations [18], graphs [19], and logic [18] [20].

For an overview of the most commonly used algorithms (Ensembles of Decision Trees (random forest (tm)[21]), Support Vector Machines (SVM) [22] [23], artificial neural networks [24] [25]) and feature selection, we refer to some excellent literature on the topic [26] [13] [15] [27].

1.3 Validation

Upon testing the reliability of a machine learning algorithm on the problem of interest—the no-free-lunch theorem [28] guarantees that no algorithm is suitable for all problems—a core assumption is that the test cases will resemble the training cases. Nearly all approaches make the assumption that examples are *independently and identically distributed* (i.i.d.). Under this assumption, the quality of the predictions can be estimated using the training data alone, using cross-validation. This means dividing it into segments (usually called folds), and then repeatedly training the model on all-but-one fold and validating it on the remaining one [15]. The repetition is done to determine the variation of the predictive performance depending on the folds. It is essential that the modeling process uses only the training data [29] and the validation on test data is done only as the final evaluation. It is a serious but unfortunately frequent methodological error to choose the algorithm parameters (and/or the learning algorithm itself) that yield the best performance on the test sets. The computed performance will then be an optimistic overestimate of the real expected performance on unseen data. A solution to select the best scoring parameters based on the training data alone is to use an internal cross-validation procedure where the training data will again be split into multiple folds. These parameters can finally be validated on the outer test fold. The whole process is then called double cross-validation. Different validation tests based on cross validation have been proposed, depending on the assumptions that can be made about the data [30] [31].

Specific performance measures can also be calculated for certain classifier algorithms such as an error bound for SVM [32] [33] [34] and out-of-bag error for Random Forests [21], which can provide alternatives to cross-validation.

For binary classifiers that only output the predicted labels, we want to assess the quality of the predictions by all appropriate metrics for its intended purpose. Examples to interpret a distribution of prediction results are accuracy, precision, false discovery rate, recall, and fall-out. The nomenclature stems from the era when book retrieval was a probabilistic process, however many synonyms, as well as additional metrics were since introduced. Accuracy is the fraction of

predictions that are correct. Precision (positive predictive value) is the accuracy of predicted positive examples, its complement is called false discovery rate. Recall (sensitivity, true positive rate) is the fraction of positive examples that are predicted positive. Fall-out (false positive rate) is the fraction of negative examples that are predicted positive.

Many classifiers can rank their predictions by confidence; they are called *ranking classifiers*. The user can then set a decision threshold, depending on the desired relative cost of the two types of errors (false positives and false negatives). Two visualization methods are commonly used to compare the quality of the rankings: *receiver operating characteristic (ROC)* and *precision-recall (PR)* curves. ROC curves relate the recall versus fall-out for the classifier operating at all decision thresholds, while PR curves focus only on positive examples. However, when comparing classifiers, the domination of one curve is equivalent to domination in the other representation [35]. To summarize performance, the area under the ROC curve (AUC-ROC) is a useful measure, which is equivalent to the probability that the classifier will score a randomly drawn positive example higher than a randomly drawn negative sample.

For any chosen performance measure, a permutation test checks if the predictions of the model are better than a random predictor. In such a test, the score of the model is compared with the distribution of scores obtained by running the model on the same data set but with the target values randomly permuted over the examples. If the score on the original data significantly deviates from this distribution, we can conclude that the data indeed contains useful structure and that the model can successfully learn from it. This permutation test is especially useful when there are few data points in a high-dimensional space [29].

2.0 Machine learning applications in proteomic analysis

This section documents the various machine learning applications that are available to support the different steps in typical shotgun and targeted proteomics experiments; we follow the steps in a typical workflow in the order in which they are usually encountered, hoping to provide an intuitive structure for researchers that are familiar with the general proteomics workflow. There are applications available supporting two-dimensional gel-electrophoresis (2-DE) image analysis [36][37][38], but they will not be discussed in detail since machine learning is only rarely used in this area.

The two data sets used to test the applications are described in the Supplementary Information.

2.1 Proteolysis

Proteolysis is an important aspect of high-throughput proteomics, since the analysis in these types of experiments is entirely centered on peptides [39]. It is therefore interesting to determine which peptides will get reliably cleaved from a specific protein, since only these peptides will have a chance to enter the LC-MS pipeline and be subsequently detected. It has long been assumed that the protease trypsin, the most frequently used cleavage workhorse in proteomics research [40], cleaves after arginine or lysine residues except when these residues are C-terminally flanked by proline, with the occasional missed cleavage occurring as well. Three distinct attempts to model the enzyme's activity more closely to the physical reality are provided by Siepen *et al.* [41], the MC:pred software [42], and the CP-DT tool [16]. In each of these cases, the objective is to improve identification rates in shotgun proteomics, and/or to provide *a priori* predictions of suitable peptides for quantitative (targeted) proteomics analyses. Predictive models have also been developed for other proteases [43], but since these focus on the biological role of the proteases, these are not discussed in detail here.

The former two predictors, Siepen *et al.* [41] and mc:pred [42] provide a score for each tryptic site that ranges from 0 to 1, with 1 meaning it will remain uncleaved. The score is constructed with the intent to remove randomly cleaved peptides from a list of potentially detectable peptides in targeted analyses, or to reduce the identification database to speed up discovery. CP-DT [16] does the inverse, and outputs a list of peptides ranked by the probability that they will occur in the specific cleaved form after tryptic proteolysis. Note that this means that CP-DT can be used not only to provide probabilities for correctly cleaved peptides, but also for peptides containing any number of missed cleavages.

All three predictors only use positional information about the amino acid sequence around the tryptic site as features in the model. No chemical information about the molecular structure of the peptide is used, nor any information about the conditions of the proteolytic digest, which can be of influence in the digestion efficiency [44].

When the performance of these algorithms is tested on publicly available data however, a difference in performance can be seen. This is not only due to the use of different underlying algorithms—an information theory based strategy by Siepen *et al.* [41], an SVM by Lawless and Hubbard [42], and an ensemble of decision trees by Fannes *et al.* [16]—but also due to differences in the filtering of the training data sets that were used in the learning stage.

We compared the performance of the three models by a ROC-curve (**Figure 2**), relating the true positive rate to the false positive rate at different thresholds of predicted cleavage probabilities. For each case (i.e. predicted correctly cleaved tryptic site) the example is considered negative if it occurs only inside an identified peptide as uncleaved, and positive if it occurs at least once at the C-terminus of an identified peptide in the test data sets. The ensemble classifier is seen to outperform the two other models in both high recall and low fall-out regimes in this analysis, offering a higher accuracy over all possible thresholds for the two data sets considered.

2.2 Liquid chromatography

Retention time prediction is one of the first problems in LC-MS to which statistical modeling was applied, and remains one of the more difficult issues. Several slightly older reviews on existing efforts are already available for this topic [45] [46], each comparing the 'gold standard' SSRCalc algorithm [47] to newer efforts. SSRCalc is based on a simple additive model that calculates retention time as a weighted sum of retention coefficients for the individual residues in a peptide and then corrects for empirical factors such as length influence and the tendency to form helical structures. Yet despite this apparent simplicity, these comparisons show that SSRCalc produces fairly accurate predictions and stacks up relatively well (albeit after inclusion of some additional correction factors such as peptide charge [48] [49]) compared to more complex analytical models [50].

Machine learning models have also been used to address the retention time prediction problem, including two artificial neural networks based models by Petritis *et al.* [51] and Shinoda *et al*. [52], and two SVM regressors, Klammer *et al.* [53], and RTPredict [54]. This latter algorithm is based on an oligo-kernel that is integrated into OpenMS [55].

Compared to these machine learning algorithms, SSRCalc has the advantage of simplicity. Since the variability in combinations of LC columns, gradients, solvents and conditions is so large, even the best model will need to be retrained or recalibrated to the conditions at hand. The more complex models suffer from the disadvantage that, while their predictive power is higher once they are correctly trained, they are slower to train and need significantly more examples to reach this power (ranging from thousands [53] to hundreds [56]). As a result, the costs of training such an algorithm, i.e. running many samples on a specific column, may well outweigh the improvements in predictive accuracy offered by the more complex algorithms.

A more recent review book chapter [57] covers the latest developments in this subdiscipline, and links to a web tool that can be used to evaluate the performance of predictors by generating scatter plots of measured retention time versus predicted retention time. Currently, the web site only offers this functionality for the authors' own predictor called rt [58], a linear model. In their book chapter, the authors correctly note that in earlier reviews there is little agreement on a metric to compare retention time predictors, and they therefore propose standard error between observed and predicted retention times as an appropriate measure. In Moruz *et al.* [8] however, where the most recent versions of SSRCalc [59] and RTPredict [60] are compared, a different metric is proposed. Moruz *et al*. use the minimum time deviation in minutes between observed and predicted retention time for 95% of the peptides \delta t_{95%}. We believe this metric to be a more useful comparator, since it provides direct information about the overall performance of the model, giving users an idea of the accuracy of the typical prediction. Note that the deviation can also be given as a percentage of the total run time rather than in minutes.

Moruz *et al.* [8] also introduce Elude, an—optionally completely re-trainable—pre-trained SVM model for peptide retention time prediction. Elude can be calibrated to a condition at hand, requiring about 200 peptides to achieve maximum accuracy, similar to what SSRCalc needs. It includes 60 features about the amino acid sequence, including hydrophobicity and helicity, but no information about charge. In the evaluation of the model, an average deviation window of 22.06% from the total length of the chromatographic run is measured for Elude, compared to 25.68% for RTPredict and 27.79% for SSRCalc. For a 90 minute run this results in an improvement of 5 minutes, a substantial gain that can significantly decrease false positive rates for identifications when used as a means to triage PSMs.

Recently, an improvement to Elude has been published, enabling it to deal with peptides bearing five kinds of posttranslational modifications that affect peptide retention time [61]. Changes to the core algorithm of Elude to allow the inclusion of this additional parameter also

led to a reduction in the deviation window for both modified and unmodified peptides, down to about 18%.

For completeness, it is worth mentioning that the Trans-Proteomic Pipeline (TPP) [62] is also able to predict retention times, using an unpublished but open source artificial neural network.

2.3 Peptide detectability

Unfortunately, not a single proteomics approach is currently able to reveal the entire population of the eluting peptides generated by proteolytic processing of a complex protein mixture. We can define peptide detectability as the probability that a given peptide will be observed in a standard sample analyzed by a standard proteomics routine [63]. Peptide detectability is determined by several factors, of which we can roughly distinguish four classes [63]: (i) the physico-chemical properties of the peptide (mass, hydrophobicity, ability to ionize or fragment); (ii) the limitations of the analytical workflow (including sample pre-processing, MS instruments and software); (iii) the abundance of the peptide in the sample; (iv) the other peptides present in the sample that compete with this peptide for subsequent detection or identification. Taken together, these factors determine whether a peptide will be detected in a particular experiment, and can also lead to variations in the detectability of peptides across experiments.

In order to predict peptide detectability, several supervised classification techniques have been applied, each attempting to model all the above factors at once. Tang *et al*. [63] used ensembles of 30 one-hidden-layer feed-forward neural networks, reporting a single accuracy estimate. Initially they identified 175 features, after which unpromising features were removed using a t-test filter. Additionally, the number of correlated features was reduced using principal

component analysis, retaining 95% variance. Later, this work was extended to include an estimation of protein abundance by using an iterative quantity adjustment [64].

Another approach was proposed by Lu *et al*. [65], [66], [67] with the APEX (Absolute Protein Expression) tool. Their peptide detectability classifier was constructed from a random forest, using feature vectors consisting of from 35 [67] up to 66 features [66]. Here again, peptide detectability prediction was used as an intermediate step in the overall task of protein quantification. After prediction of peptide detectability, protein quantification was performed by normalizing the observed spectral count by the predicted count.

The approach by Mallick *et al*. [68] predicts whether or not peptides are proteotypic. They provided several distinct predictors, each specific to a particular experimental design. Mallick *et al*. identified 494 amino acid features, which were subsequently summed and averaged for each peptide, resulting in almost 1,000 distinct features. Next, based on the Kullbach-Leibler distance and the Kolmogorov-Smirnov distance, the smallest descriptive subset of properties was determined. Based on this subset, a Gaussian mixture discriminant function was developed from the training data. This application is named PeptideSieve.

Similar to the approach of Tang *et al*., Sanders *et al*. employ a neural network as binary classifier, called PepFly [69]. However, they do not provide a single general classifier, instead they mention that the classifier should be retrained for each distinct experimental setup. Sanders *et al*. identified 596 features, of which a reduced set is calculated using a greedy search through feature space, after which the selected features are used to construct a one-hidden-layer feed-forward neural network.

Wedge *et al*. on the other hand use a genetic programming classifier [70]. They selected 393 peptide properties for their initial genetic program. Afterwards, based on the usage of input nodes in this initial program, the set of input nodes was decreased to 34 and 6 input nodes.

ESPPredictor by Fusaro *et al*. is based on a random forest to predict high-responding peptides [71]. They identified 550 physico-chemical properties as features. In contrast to the

other approaches however, no feature selection is performed here. Instead, all features are used to construct a random forest consisting of 50 000 trees. The output is a probabilistic value, corresponding to the fraction of trees that predict the peptide to be detectable.

Next, the predictor by Webb-Robertson *et al*., named STEPP (SVM Technique for Evaluating Proteotypic Peptides) uses a support vector machine with a quadratic kernel [72]. Out of 35 initial features, feature selection is performed based on the Fisher Criterion Score.

Finally, Eyers *et al*. combined several of the above mentioned types of classifiers in order to create a consensus algorithm named CONSeQuence [73]. Concretely, a random forest, a genetic program, an artificial neural network, and a support vector machine are each individually used. Afterwards, a specific number of votes from each of the four predictors is required in order to classify a peptide as detectable or not.

As shown above, several algorithms have thus been used to solve the peptide detectability problem, with all of them relying on a different range of features. However, appropriately training the classifier plays a crucial role in obtaining a good performance. Particularly, a classifier trained on a data set pertaining to a specific experimental setup, will only perform optimally for the same experimental setup. Indeed, Mueller *et al.* [74] showed that the PeptideSieve predictor by Mallick *et al*. [68] was less accurate when applied to a different data set. In addition to training the model on an appropriate data set, a careful selection of the training data is also required, i.e., obtaining a balanced set of positive and negative examples.

We tested the pre-trained classifiers, compatible with the two data sets described in the supplementary information: Peptide Detectability Predictor (http://darwin.informatics.indiana.edu/applications/PeptideDetectabilityPredictor/), based on the work of Tang *et al*. [63]; PeptideSieve (http://tools.proteomecenter.org/wiki/index.php?title=Software:PeptideSieve), by Mallick *et al*. [68]; ESPPredictor (http://www.broadinstitute.org/cancer/software/genepattern/modules/ESPPredictor.html), by

Fusaro *et al*. [71]; STEPP (http://www.biopilot.org/portal/software/stepp.html), by Webb-Robertson *et al*. [72]; and CONSeQuence (http://king.smith.man.ac.uk/CONSeQuence/), by Eyers *et al*. [73].

We compared their performance in predicting the actually detected peptides out of a list of peptides generated by an *in silico* cleavage of the data set by the Keil rules [75]. That way we are able to compare classifier software (Peptide Detectability Predictor, PeptideSieve and CONSeQuence) that take proteins as input and perform cleavage themselves. Furthermore, to ensure a fair comparison, for peptide generation, no uncleaved sites were allowed; Peptide Detectability Predictor does not, unlike all other predictors, offer the possibility to set the amount of uncleaved sites. Several potential peptide candidates are thus not included in the analysis. However, the mutual differences obtained between the various predictors should remain valid. In addition to the previous parameters, the minimal peptide sequence length was set to five, PeptideSieve was run using the PAGE_ESI experimental design choice, and the rank score prediction type for CONSeQuence was used.

An example peptide is considered positive if the predicted peptide was actually observed in the data set. Likewise, an example is considered negative if it was not observed in the data set. The ROC-analysis in **Figure 3** shows that the predictors Peptide Detectability Predictor, PeptideSieve, and CONSeQuence perform very similarly over the two data sets, while ESPPredictor seems to be the best choice when low fall-out is desired, and STEPP appears to be the best choice when high recall is desired.

2.4 Peptide identification

To match MS/MS spectra to the peptides that generated them, database search methods, known as search engines, constitute by far the most popular approaches. The

databases that are searched in these approaches can either contain the peptide sequences expected to be observed in the sample [76], or (processed) MS/MS spectra that have been observed and identified in previous experiments [77]. In both cases a measure of similarity is required that scores candidate peptides against an experimental MS/MS spectrum. However, the best scoring peptide-spectrum-match (PSM) can still be wrong. The challenge is to separate the correct PSMs from the incorrect ones, which is a typical machine learning task.A first attempt to find a good separation between correct and incorrect PSMs is to cluster the matching score distribution with a probabilistic mixture modeling method [78]. This clustering approach was further improved by representing PSMs as complex feature vectors that contain more information about the proteomics experiment such as the expected number of matches from a given database, the effective database size, a correction for indistinguishable peptides, or a measurement of match quality [79–83]. This algorithm is integrated in the Trans Proteomic Pipeline under the name PeptideProphet [62].

To solve the match separation challenge as a supervised learning task one requires examples of both correct and incorrect PSMs. Such a data set can be obtained from purified proteome samples. In this case the expected peptide identifications are considered true matches, whereas unexpected identifications are considered false matches. These rather limited data sets [78] [84] allowed for supervised methods to compute accurate classifiers (as compared to the clustering approach) using support vector machines [85] [86] [87] , random forests [85], and neural networks [88] [89].

Examples of correct and incorrect PSMscan also be obtained from the actual proteomics experiment itself. By searching MS/MS spectra against both a target as well as a decoy database [90] one obtains examples of incorrect PSMsfrom the decoy database. Matches against the target database constitute both correct and incorrect matches. A semi-supervised support vector machine is shown to learn accurate PSM classifiers from this data using an iterative support vector machine learning procedure, initially identifying a small set of high-

scoring target PSMs, and then learning to separate these from the decoy PSMs. The learned classifier is applied to re-score the entire set, and if new high-scoring PSMs are identified, then the procedure is repeated; this algorithm is called Percolator [9,91][92]. By using a special loss function this task can be solved more accurately using a supervised learning algorithm. This loss function does not severely penalize examples that are far from the decision boundary such that accurate non-linear support vector machine classifiers can be learned, even though incorrect target PSMs labeled as correct are present in the dataset [93].

A similar approach where PSMs are represented by feature vectors can be used to combine the results of different search engines to obtain more accurate PSM scores[94]. The best results are obtained by computing complex features from the observations made by all search engines combined and applying linear discriminant analysis (iProphet [83]) or random forest learning (PepArML [95]) in a similar semi-supervised way as described above.

2.5 Fragmentation

The fragmentation of a peptide using methods like collision induced dissociation (CID) produces signal spectra that contain information about the chemical dissociation pattern of the fragmented peptide. The signal peaks in an MS/MS spectrum indicate the presence of a peptide fragment ion with a specific mass, and the intensity of such a peak is dependent on a number of factors such as the abundance of the peptide in the sample and the efficiency of the bond breaking that generated the fragment. Also at play are the proteotypicity of the fragment ion, as well as factors related to the peptide and the instrument that generated the MS/MS spectrum [96]. Being able to predict signal peak intensities is important for the understanding of the patterns behind peptide fragmentation.

Elias *et al.* implemented an inductive Bayesian decision tree approach to model peptide fragmentation and showed that a decision tree model representation is highly suitable for learning the diverse set of rules that govern peptide fragmentation [97]. Their data-driven

approach was able to extract, from 27 000 peptide-spectrum matches, many of the known fragmentation rules and discovered several new ones. However, their approach does not model the peak intensities directly. Rather it models the probability of observing a certain fragment ion intensity bin. A similar study based on Bayesian neural networks was presented in [98] with a data set of 13 900 peptide-spectrum matches.

Approaches that model peak intensities directly exist as well. PeptideART [99] implements an ensemble of neural networks that each model the most important fragment ion peak intensities in a multi-output feed-forward one-hidden-layer neural network. The features used as input to the neural network are very similar to those suggested by Elias *et al.* The authors reported a systematic assessment of the accuracy of the current peptide MS/MS spectrum predictors for the most commonly used CID instruments [100]. They found that PeptideART achieves generally higher accuracy on a wide range of proteomic data sets when trained on a data set of 41 054 peptide-spectrum matches.

Another promising approach is to predict the intensity ranks of the fragment ions in an MS/MS spectrum. Frank *et al.* implemented a discriminative ranking-based model that applies boosting to model the relationship between simple sequence-based features and the observed peak intensity ranks [101].

A recent approach predicts an entire fragmentation spectrum at once based on a weighted K-nearest neighbors algorithm applied to a spectral library [102]. The distance function used by the nearest neighbor method is learned with an SVM, and the method is used to complete a spectral library and improve results of search engines.

2.6 Protein inference

In a peptide-centric proteomic approach, proteins are identified based on peptide homology [103]. Previous statements accentuate that proteins are not analyzed directly by mass

spectrometry, but by partial analysis of fragment ions of their peptides, which are obtained after proteolysis and separation *via* liquid chromatography. The connectivity between proteins and peptides is thus lost in various ways during these steps in the overall analytical process. Protein inference is then the reassembly process that identifies and characterizes (including PTMs) the proteins in the sample based on the observed evidence, i.e., the identified peptides. The challenge in protein inference is to correctly identify these precursor protein sequences, since a limited set of peptides may be assigned to multiple proteins. The protein inference problem is well described in the literature [104] [10,105,106].

It is worth mentioning that the main effort in protein inference is not related to spectral matching as described in the section on peptide identification, but on the induction of protein to peptide connectivity after peptide identification. Therefore, mostly information about peptide identification and protein accession numbers is used to solve the inference problem, as depicted by the emphasized bipartite graph in **Figure 4**. A trivial solution to this problem could be the most complete set of proteins that correspond to the observed peptides. In contrast, a more reductionist approach is to adopt the principle of Occam's razor and to report the minimal sets of proteins that could explain the observed peptides. However, it is not likely that previous solutions entirely reflect the underlying natural process, and probably more evidence can be incorporated in the computational approach towards more directed protein inference. For example, auxiliary and derivative information from the peptide-to-spectrum matches , e.g., delta mass, missed cleavage, charge, competing peptides for the same spectrum, etc., could gain support for a particular protein, while information about the spectral data layer is often ignored in current protein inference algorithms. For completeness, we graphically illustrate the inference problem in **Figure 4** with its available meta-information. It is natural to represent the inference problem as a probabilistic relational model [18] [107]. From the figure it is clear that there are three layers of information. Each layer represents a step in the inference process. The nodes in

the layers represent: (i) spectral data: m/z, RT, intensity, total ion count, etc; (ii) peptide sequence annotation: mass, calculated pI, predicted fragment ions, uncleaved sites, etc; and (iii) protein information: mass, number of peptides, sequence length, PTMs, protein group, etc.

The edges between the layers also contain information that can be used for protein inference: edge 1-2, PSM: number of fragment ions matched, PSM score, p-value, e-value, rank, etc.; and edge 2-3, Peptide-Protein Connection: parsimony, proteotypic peptides, peptide observability, intensity-abundance relation, etc.

To determine a minimum set of peptides that can be uniquely assigned to proteins, the one- or two-peptide rules on only a small set of highly confident first ranked peptides are often employed[108]. These simple rules however only allow for identification of a very small portion of the proteins in a sample. Most modern inference methods try to incorporate more of the available information in combination with a few heuristic rules. *Sensu stricto*, these heuristic methods cannot be classified as machine learning methods [109] [110] [111] [112][113][114] [115] [116][117]. We review a set of implementations rooted in machine learning below.

Usually methods inferring a minimal set of proteins that covers the most likely and confidently identifiable peptides are restricted to the bipartite graph [118] [119] [120] as depicted in **Figure 4**, but Spivak *et al*. recently [121] proposed a single optimization problem that also incorporates information about the spectral layer. The approach was motivated by the observation that the peptide and protein level tasks are cooperative, and the solution to each can be improved by using information about the solution to the other. This relation is illustrated by a feedback edge that connects the protein layer with the spectral layer in **Figure 4**. The approach is implemented in the Barista-tool of the Crux toolkit [122] and relies on an artificial neural network to solve the optimization problem.

Other methods, borrowed from language modeling can be used to for the inference problem as well [123]. Yang *et al.* [124] apply a vector space model, used in information retrieval to model document identifiers, to model peptides as protein identifiers.

An interesting development is the use of peptide detectability in the inference problem [63] [125] [64], for which the above section covers the available algorithms. In this context, the graph representation of the protein inference problem can be modified by adding probabilities to a peptide-protein connection. Doing so allows information about detectable but unidentified peptides to be included in the inference.

Another development is the use of information about peptide intensity and protein abundance. This approach was first described to solve the shared peptide problem in protein quantification [126], however, it allows for pruning the degenerate peptides as well. As such it can be considered that the protein inference problem is generalized by the protein quantification problem [127] [126] [97].

The plethora of proposed solutions performing protein inference [10]serves as a good illustration of the magnitude of the problem. Validation of protein inference and quantification is however still a topic of ongoing research [128].

3.0 Discussion

While we can conclude that there are already many methods available for most of the steps in a typical proteomics workflow, a general remark that can be brought up is that the diversity in interfaces offered by these tools, and the variety in support by workflow managers

for these tools is simply too large. Software that implements the algorithms is unfortunately not always available, or is often provided without support for the input or output of standardized data formats. This lack of end-user oriented detail leads to a limited uptake in the field, except for those tools that are integrated in popular pipelines such as the TPP, or that are embedded in commercial software.

The field would also greatly benefit from the availability of standardized data sets for testing and evaluation purposes, especially if matched to unified testing measures that can allow benchmarking of novel algorithms or software tools. As for the availability of test data sets, in most cases the data originally used for testing algorithm performance are not made available, and even if a public data set is used, it is usually filtered. This can result in wildly varying performance measurements depending on the stability of the method. The availability of heterogeneous but documented testing data [129] [130] [131] however only starts to address this issue. Indeed, as pointed out above, the filtered data should be easy to obtain as well.

As for the performance measurement itself, a universally applicable and standardized test *per* application in a workflow,, could greatly improve transparency in the field. Firstly, it would mean that all software should communicate in open, standardized formats (HUPO-PSI [132]), as for example all OpenMS tools [55] already do. Secondly, the process for testing should be transparently available, so measures applicable to any of these software tools can be easily created, applied and compared.

We believe that moving forward as a field can be achieved by enabling researchers to publish predictions on individual instances of the available data sets. This will enable objective comparison of predictors, on instances tailor-made to one proteomics application. In general, the more is published about the data, predictions or algorithms, the more reproducible and easier the comparison will be. A possible measure to facilitate the comparison over multiple predictors is the establishment of an online machine learning platform for proteomics research. Such a portal would host public data instances along with published predictor software, and

automatically evaluate and visualize the performance of the submitted predictors through a transparent (open-source) process. Simply the availability of an open-source process to evaluate the predictor software would already help potential users out to set up the software themselves.

Examples of such online platforms in other fields where machine learning developers can submit tools and essentially enter a competition on building the best model are KKD with the KKD Cup (www.sigkdd.org/kddcup/index.php), Challenges in Machine Learning (www.chalearn.org/challenges.html), Tuned IT Data Competitions (http://tunedit.org), OpenML (http://expdb.cs.kuleuven.be), and (http://mldata.org), a machine learning data set repository. There is even an example of commercial success with such a platform: http://www.kaggle.com/competitions [133].

Machine learners, who are often specialized in a subset of algorithms, could improve their models simply by employing additional relevant feature sets that are published in different models, as is evident from the analysis of the models above. In doing so, they could leverage recent advances in learning from richer representations such as statistical relational models or learning from graphs. Models can also be chained, for example the union of a retention time predictor and a peptide detectability predictor could filter for false positive identifications more stringently than any of these two predictors could achieve separately. As for peptide detectability predictors, we believe that instead of modeling the whole process from peptides (or even proteins) to the detector as a single, large black box, the combination of separate models for e.g. ionizability and competition with other peptides in the sample, could result in more flexible and instrument-independent models as well a better understanding of the processes involved.

Overall, however, it may be clear that machine learning applications are already providing solid informatics support for many problems in proteomics. Yet despite the plethora of tools already published, there still remains substantial room for improvements or additions to this repertoire.

With the availability of increasing amounts of publicly available, high-quality data, it should become much more straightforward for interested data scientists to start tackling such issues, and it is therefore not unrealistic to expect that the intensity with which machine learning will be applied to solve problems in proteomics will only increase in the foreseeable future.

Acknowledgments

This work was primarily supported by SBO grant 'InSPECtor' (120025) of the Flemish agency for Innovation by Science and Technology (IWT). H.B. is supported by the Research Council of Norway. L.M. acknowledges the support of Ghent University (Multidisciplinary Research Partnership “Bioinformatics: from nucleotides to networks”), the PRIME-XS project, grant agreement number 262067, and the 'ProteomeXchange' project, grant agreement number 260558, both funded by the European Union 7th Framework Program. J.R. acknowledges the support by ERC grant no. 240186 "MiGraNT".

The authors declare no conflict of interest.

References

[1] Hooda, Y., Kim, P.M., Computational structural analysis of protein interactions and networks. *PROTEOMICS* 2012, 12, 1697–1705.

[2] Srihari, S., Leong, H.W., A survey of computational methods for protein complex prediction from protein interaction networks. *J. Bioinform. Comput. Biol.* 2013, 11, 1230002.

[3] Elo, L.L., Schwikowski, B., Mining proteomic data for biomedical research. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 2012, 2, 1–13.

[4] Wu, B., Abbott, T., Fishman, D., McMurray, W., et al., Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data. *Bioinformatics* 2003, 19, 1636–1643.

[5] Qu, Y., Adam, B.-L., Yasui, Y., Ward, M.D., et al., Boosted Decision Tree Analysis of Surface-enhanced Laser Desorption/Ionization Mass Spectral Serum Profiles Discriminates Prostate Cancer from Noncancer Patients. *Clin. Chem.* 2002, 48, 1835–1843.

[6] Listgarten, J., Emili, A., Statistical and Computational Methods for Comparative Proteomic Profiling Using Liquid Chromatography-Tandem Mass Spectrometry. *Mol. Cell. Proteomics* 2005, 4, 419–434.

[7] Geurts, P., Fillet, M., Seny, D. de, Meuwis, M.-A., et al., Proteomic mass spectra classification using decision tree based ensemble methods. *Bioinformatics* 2005, 21, 3138–3145.

[8] Moruz, L., Tomazela, D., Käll, L., Training, Selection, and Robust Calibration of Retention Time Models for Targeted Proteomics. *J. Proteome Res.* 2010, 9, 5209–5216.

[9] Wright, J.C., Collins, M.O., Yu, L., Kall, L., et al., Enhanced Peptide Identification by Electron Transfer Dissociation Using an Improved Mascot Percolator. *Mol. Cell. Proteomics* 2012, 11, 478–491.

[10] Huang, T., Wang, J., Yu, W., He, Z., Protein inference: a review. *Brief. Bioinform.* 2012, 13, 586–614.

[11] Claassen, M., Aebersold, R., Buhmann, J.M., Proteome Coverage Prediction for Integrated Proteomics Datasets. *J. Comput. Biol.* 2011, 18, 283–293.

[12] Ridder, D. de, Ridder, J. de, Reinders, M.J.T., Pattern recognition in bioinformatics. *Brief. Bioinform.* 2013.

[13] Witten, I.H., *Data mining: practical machine learning tools and techniques*, Morgan Kaufmann, Burlington, MA 2011.

[14] Bishop, C.M., *Pattern recognition and machine learning*, Springer, New York 2006.

[15] Blockeel, H., *Machine Learning and Inductive Inference*, ACCO Leuven, 2010.

[16] Fannes, T., Vandermarliere, E., Schietgat, L., Degroeve, S., et al., Predicting Tryptic Cleavage from Proteomics Data Using Decision Tree Ensembles. *J. Proteome Res.* 2013.

[17] Käll, L., Canterbury, J.D., Weston, J., Noble, W.S., MacCoss, M.J., Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nat. Methods* 2007, 4, 923–925.

[18] De Raedt, L., *Logical and relational learning*, Springer, New York 2008.

[19] Borgwardt, K.M., Graph Kernels. Text.PhDThesis. lmu, 2007.
[20] Frasconi, P., Costa, F., De Raedt, L., De Grave, K., *kLog: A Language for Logical and Relational Learning with Kernels*, 2012.
[21] Breiman, L., Random Forests. *Mach. Learn.* 2001, 45, 5–32.
[22] Cortes, C., Vapnik, V., Support-vector networks. *Mach. Learn.* 1995, 20, 273–297.
[23] Cristianini, N., *An introduction to support vector machines: and other kernel-based learning methods*, Cambridge University Press, Cambridge□; New York 2000.
[24] Bishop, C.M., *Neural Networks for Pattern Recognition*, Oxford University Press, Inc., New York, NY, USA 1995.
[25] Haykin, S., *Neural Networks: A Comprehensive Foundation*, Prentice Hall PTR, Upper Saddle River, NJ, USA 1998.
[26] Das, S., in:, *Mach. Learn.-Int. Work. THEN Conf.-*, 2001, pp. 74–81.
[27] Saeys, Y., Inza, I., Larrañaga, P., A review of feature selection techniques in bioinformatics. *Bioinformatics* 2007, 23, 2507–2517.
[28] Wolpert, D.H., The Lack of A Priori Distinctions Between Learning Algorithms. *Neural Comput.* 1996, 8, 1341–1390.
[29] Westerhuis, J.A., Hoefsloot, H.C.J., Smit, S., Vis, D.J., et al., Assessment of PLSDA cross validation. *Metabolomics* 2008, 4, 81–89.
[30] Dietterich, T.G., Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Comput.* 1998, 10, 1895–1923.
[31] Granholm, V., Noble, W.S., Kall, L., A cross-validation scheme for machine learning algorithms in shotgun proteomics. *BMC Bioinformatics* 2012, 13, S3.
[32] Vapnik, V., Chapelle, O., Bounds on error expectation for support vector machines. *Neural Comput.* 2000, 12, 2013–2036.
[33] Chung, K.-M., Kao, W.-C., Sun, C.-L., Wang, L.-L., Lin, C.-J., Radius Margin Bounds for Support Vector Machines with the RBF Kernel. *Neural Comput.* 2003, 15, 2643–2681.
[34] Li, H., Wang, S., Qi, F., in:, Zhang J, He J-H, Fu Y (Eds.), *Comput. Inf. Sci.*, Springer Berlin Heidelberg, 2005, pp. 1067–1071.
[35] Davis, J., Goadrich, M., in:, *Proc. 23rd Int. Conf. Mach. Learn.*, ACM, New York, NY, USA 2006, pp. 233–240.
[36] Dowsey, A.W., English, J.A., Lisacek, F., Morris, J.S., et al., Image analysis tools and emerging algorithms for expression proteomics. *PROTEOMICS* 2010, 10, 4226–4257.
[37] Tsakanikas, P., Manolakos, E.S., Protein spot detection and quantification in 2-DE gel images using machine-learning methods. *PROTEOMICS* 2011, 11, 2038–2050.
[38] Savelonas, M.A., Mylona, E.A., Maroulis, D., Unsupervised 2D gel electrophoresis image segmentation based on active contours. *Pattern Recognit.* 2012, 45, 720–731.
[39] Gevaert, K., Van Damme, P., Ghesquière, B., Impens, F., et al., A la carte proteomics with an emphasis on gel-free techniques. *PROTEOMICS* 2007, 7, 2698–2718.
[40] Vandermarliere, E., Mueller, M., Martens, L., Getting intimate with trypsin, the leading protease in proteomics. *Mass Spectrom. Rev.* 2013, 000–000.

[41] Siepen, J.A., Keevil, E.-J., Knight, D., Hubbard, S.J., Prediction of missed cleavage sites in tryptic peptides aids protein identification in proteomics. *J. Proteome Res.* 2007, 6, 399–408.
[42] Lawless, C., Hubbard, S.J., Prediction of missed proteolytic cleavages for the selection of surrogate peptides for quantitative proteomics. *Omics J. Integr. Biol.* 2012, 16, 449–456.
[43] Verspurten, J., Gevaert, K., Declercq, W., Vandenabeele, P., SitePredicting the cleavage of proteinase substrates. *Trends Biochem. Sci.* 2009, 34, 319–323.
[44] Brownridge, P., Beynon, R.J., The importance of the digest: Proteolysis and absolute quantification in proteomics. *Methods* 2011, 54, 351–360.
[45] Bączek, T., Kaliszan, R., Predictions of peptides' retention times in reversed-phase liquid chromatography as a new supportive tool to improve protein identification in proteomics. *PROTEOMICS* 2009, 9, 835–847.
[46] Babushok, V.I., Zenkevich, I.G., Retention Characteristics of Peptides in RP-LC: Peptide Retention Prediction. *Chroma* 2010, 72, 781–797.
[47] Krokhin, O.V., Sequence-Specific Retention Calculator. Algorithm for Peptide Retention Prediction in Ion-Pair RP-HPLC: Application to 300- and 100-Å Pore Size C18 Sorbents. *Anal. Chem.* 2006, 78, 7785–7795.
[48] Vu, H., Spicer, V., Gotfrid, A., Krokhin, O.V., A model for predicting slopes S in the basic equation for the linear-solvent-strength theory of peptide separation by reversed-phase high-performance liquid chromatography. *J. Chromatogr. A* 2010, 1217, 489–497.
[49] Spicer, V., Grigoryan, M., Gotfrid, A., Standing, K.G., Krokhin, O.V., Predicting Retention Time Shifts Associated with Variation of the Gradient Slope in Peptide RP-HPLC. *Anal. Chem.* 2010, 82, 9678–9685.
[50] Perlova, T.Y., Goloborodko, A.A., Margolin, Y., Pridatchenko, M.L., et al., Retention time prediction using the model of liquid chromatography of biomacromolecules at critical conditions in LC-MS phosphopeptide analysis. *PROTEOMICS* 2010, 10, 3458–3468.
[51] Petritis, K., Kangas, L.J., Yan, B., Strittmatter, E.F., et al., Improved peptide elution time prediction for reversed-phase liquid chromatography-MS by incorporating peptide sequence information. *Anal. Chem.* 2006, 78, 5026–5039.
[52] Shinoda, K., Sugimoto, M., Yachie, N., Sugiyama, N., et al., Prediction of Liquid Chromatographic Retention Times of Peptides Generated by Protease Digestion of the Escherichia coli Proteome Using Artificial Neural Networks. *J. Proteome Res.* 2006, 5, 3312–3317.
[53] Klammer, A.A., Yi, X., MacCoss, M.J., Noble, W.S., Improving Tandem Mass Spectrum Identification Using Peptide Retention Time Prediction across Diverse Chromatography Conditions. *Anal. Chem.* 2007, 79, 6111–6118.
[54] Pfeifer, N., Leinenbach, A., Huber, C.G., Kohlbacher, O., Statistical learning of peptide retention behavior in chromatographic separations: a new kernel-based approach for computational proteomics. *BMC Bioinformatics* 2007, 8, 468.
[55] Sturm, M., Bertsch, A., Gröpl, C., Hildebrandt, A., et al., OpenMS – An open-source software framework for mass spectrometry. *BMC Bioinformatics* 2008, 9, 163.

[56] Shamshurin, D., Spicer, V., Krokhin, O.V., Defining intrinsic hydrophobicity of amino acids' side chains in random coil conformation. Reversed-phase liquid chromatography of designed synthetic peptides vs. random peptide data sets. *J. Chromatogr. A* 2011, 1218, 6348–6355.

[57] Henneman, A.A., Palmblad, M., in:, Matthiesen R (Ed.), *Mass Spectrom. Data Anal. Proteomics*, Humana Press, 2013, pp. 101–118.

[58] Palmblad, M., Ramström, M., Markides, K.E., Håkansson, P., Bergquist, J., Prediction of Chromatographic Retention and Protein Identification in Liquid Chromatography/Mass Spectrometry. *Anal. Chem.* 2002, 74, 5826–5830.

[59] Krokhin, O.V., Spicer, V., Peptide Retention Standards and Hydrophobicity Indexes in Reversed-Phase High-Performance Liquid Chromatography of Peptides. *Anal. Chem.* 2009, 81, 9522–9530.

[60] Pfeifer, N., Leinenbach, A., Huber, C.G., Kohlbacher, O., Improving peptide identification in proteome analysis by a two-dimensional retention time filtering approach. *J. Proteome Res.* 2009, 8, 4109–4115.

[61] Moruz, L., Staes, A., Foster, J.M., Hatzou, M., et al., Chromatographic retention time prediction for posttranslationally modified peptides. *PROTEOMICS* 2012, 12, 1151–1159.

[62] Deutsch, E.W., Mendoza, L., Shteynberg, D., Farrah, T., et al., A Guided Tour of the Trans-Proteomic Pipeline. *Proteomics* 2010, 10, 1150–1159.

[63] Tang, H., Arnold, R.J., Alves, P., Xun, Z., et al., A computational approach toward label-free protein quantification using predicted peptide detectability. *Bioinformatics* 2006, 22, e481–e488.

[64] Li, Y.F., Arnold, R.J., Tang, H., Radivojac, P., The importance of peptide detectability for protein identification, quantification, and experiment design in MS/MS proteomics. *J. Proteome Res.* 2010, 9, 6288–6297.

[65] Lu, P., Vogel, C., Wang, R., Yao, X., Marcotte, E.M., Absolute protein expression profiling estimates the relative contributions of transcriptional and translational regulation. *Nat. Biotechnol.* 2007, 25, 117–124.

[66] Vogel, C., Marcotte, E.M., Calculating absolute and relative protein abundance from mass spectrometry-based protein expression data. *Nat. Protoc.* 2008, 3, 1444–1451.

[67] Braisted, J.C., Kuntumalla, S., Vogel, C., Marcotte, E.M., et al., The APEX Quantitative Proteomics Tool: generating protein quantitation estimates from LC-MS/MS proteomics results. *BMC Bioinformatics* 2008, 9, 529.

[68] Mallick, P., Schirle, M., Chen, S.S., Flory, M.R., et al., Computational prediction of proteotypic peptides for quantitative proteomics. *Nat. Biotechnol.* 2007, 25, 125–131.

[69] Sanders, W.S., Bridges, S.M., McCarthy, F.M., Nanduri, B., Burgess, S.C., Prediction of peptides observable by mass spectrometry applied at the experimental set level. *BMC Bioinformatics* 2007, 8 Suppl 7, S23.

[70] Wedge, D.C., Gaskell, S.J., Hubbard, S.J., Kell, D.B., et al., in:, *Proc. 9th Annu. Conf. Genet. Evol. Comput.*, ACM, New York, NY, USA 2007, pp. 2219–2225.

[71] Fusaro, V.A., Mani, D.R., Mesirov, J.P., Carr, S.A., Prediction of high-responding peptides for targeted protein assays by mass spectrometry. *Nat. Biotechnol.* 2009, 27, 190–198.
[72] Webb-Robertson, B.-J.M., Cannon, W.R., Oehmen, C.S., Shah, A.R., et al., A support vector machine model for the prediction of proteotypic peptides for accurate mass and time proteomics. *Bioinformatics* 2010, 26, 1677–1683.
[73] Eyers, C.E., Lawless, C., Wedge, D.C., Lau, K.W., et al., CONSeQuence: Prediction of Reference Peptides for Absolute Quantitative Proteomics Using Consensus Machine Learning Approaches. *Mol. Cell. Proteomics MCP* 2011, 10.
[74] Mueller, M., Vizcaíno, J.A., Jones, P., Côté, R., et al., Analysis of the experimental detection of central nervous system-related genes in human brain and cerebrospinal fluid datasets. *Proteomics* 2008, 8, 1138–1148.
[75] Keil, P.D.B., *Specificity of Proteolysis*, Springer Berlin Heidelberg, 1992.
[76] Sadygov, R.G., Cociorva, D., Yates, J.R., 3rd, Large-scale database searching using tandem mass spectra: looking up the answer in the back of the book. *Nat. Methods* 2004, 1, 195–202.
[77] Lam, H., Deutsch, E.W., Eddes, J.S., Eng, J.K., et al., Building consensus spectral libraries for peptide identification in proteomics. *Nat. Methods* 2008, 5, 873–875.
[78] Keller, A., Nesvizhskii, A.I., Kolker, E., Aebersold, R., Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Anal. Chem.* 2002, 74, 5383–5392.
[79] Moore, R.E., Young, M.K., Lee, T.D., Qscore: an algorithm for evaluating SEQUEST database search results. *J. Am. Soc. Mass Spectrom.* 2002, 13, 378–386.
[80] Razumovskaya, J., Olman, V., Xu, D., Uberbacher, E.C., et al., A computational method for assessing peptide- identification reliability in tandem mass spectrometry analysis with SEQUEST. *Proteomics* 2004, 4, 961–969.
[81] López-Ferrer, D., Martínez-Bartolomé, S., Villar, M., Campillos, M., et al., Statistical model for large-scale peptide identification in databases from tandem mass spectra using SEQUEST. *Anal. Chem.* 2004, 76, 6853–6860.
[82] Sun, W., Li, F., Wang, J., Zheng, D., Gao, Y., AMASS: software for automatically validating the quality of MS/MS spectrum from SEQUEST results. *Mol. Cell. Proteomics MCP* 2004, 3, 1194–1199.
[83] Shteynberg, D., Deutsch, E.W., Lam, H., Eng, J.K., et al., iProphet: multi-level integrative analysis of shotgun proteomic data improves peptide and protein identification rates and error estimates. *Mol. Cell. Proteomics MCP* 2011, 10, M111.007690.
[84] Keller, A., Purvine, S., Nesvizhskii, A.I., Stolyar, S., et al., Experimental protein mixture for validating tandem mass spectral analysis. *Omics J. Integr. Biol.* 2002, 6, 207–212.
[85] Ulintz, P.J., Zhu, J., Qin, Z.S., Andrews, P.C., Improved classification of mass spectrometry database search results using newer machine learning approaches. *Mol. Cell. Proteomics MCP* 2006, 5, 497–509.

[86] Anderson, D.C., Li, W., Payan, D.G., Noble, W.S., A new algorithm for the evaluation of shotgun peptide sequencing in proteomics: support vector machine classification of peptide MS/MS spectra and SEQUEST scores. *J. Proteome Res.* 2003, 2, 137–146.
[87] Wang, H., Fu, Y., Sun, R., He, S., et al., An SVM scorer for more sensitive and reliable peptide identification via tandem mass spectrometry. *Pac. Symp. Biocomput. Pac. Symp. Biocomput.* 2006, 303–314.
[88] Razumovskaya, J., Olman, V., Xu, D., Uberbacher, E.C., et al., A computational method for assessing peptide- identification reliability in tandem mass spectrometry analysis with SEQUEST. *Proteomics* 2004, 4, 961–969.
[89] Baczek, T., Buciński, A., Ivanov, A.R., Kaliszan, R., Artificial neural network analysis for evaluation of peptide MS/MS spectra in proteomics. *Anal. Chem.* 2004, 76, 1726–1732.
[90] Elias, J.E., Gygi, S.P., Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nat. Methods* 2007, 4, 207–214.
[91] Käll, L., Canterbury, J.D., Weston, J., Noble, W.S., MacCoss, M.J., Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nat. Methods* 2007, 4, 923–925.
[92] Ding, Y., Choi, H., Nesvizhskii, A.I., Adaptive Discriminant Function Analysis and Re-ranking of MS/MS Database Search Results for Improved Peptide Identification in Shotgun Proteomics. *J. Proteome Res.* 2008, 7, 4878–4889.
[93] Spivak, M., Weston, J., Bottou, L., Kall, L., Noble, W.S., Improvements to the Percolator algorithm for peptide identification from shotgun proteomics data sets. *J. Proteome Res.* 2009, 8, 3737–3745.
[94] Shteynberg, D., Nesvizhskii, A.I., Moritz, R.L., Deutsch, E.W., Combining Results of Multiple Search Engines in Proteomics. *Mol. Cell. Proteomics* 2013, 12, 2383–2393.
[95] Edwards, N., Wu, X., Tseng, C.-W., An Unsupervised, Model-Free, Machine-Learning Combiner for Peptide Identifications from Tandem Mass Spectra. *Clin. Proteomics* 2009, 5, 23–36.
[96] Barton, S.J., Whittaker, J.C., Review of factors that influence the abundance of ions produced in a tandem mass spectrometer and statistical methods for discovering these factors. *Mass Spectrom. Rev.* 2009, 28, 177–187.
[97] Elias, J.E., Gibbons, F.D., King, O.D., Roth, F.P., Gygi, S.P., Intensity-based protein identification by machine learning from a library of tandem mass spectra. *Nat. Biotechnol.* 2004, 22, 214–219.
[98] Zhou, C., Bowler, L.D., Feng, J., A machine learning approach to explore the spectra intensity pattern of peptides using tandem mass spectrometry data. *BMC Bioinformatics* 2008, 9, 325.
[99] Arnold, R.J., Jayasankar, N., Aggarwal, D., Tang, H., Radivojac, P., A machine learning approach to predicting peptide fragmentation spectra. *Pac. Symp. Biocomput. Pac. Symp. Biocomput.* 2006, 219–230.

[100] Li, S., Arnold, R.J., Tang, H., Radivojac, P., On the accuracy and limits of peptide fragmentation spectrum prediction. *Anal. Chem.* 2011, 83, 790–796.
[101] Frank, A.M., Predicting Intensity Ranks of Peptide Fragment Ions. *J. Proteome Res.* 2009, 8, 2226–2240.
[102] Ji, C., Arnold, R.J., Sokoloski, K.J., Hardy, R.W., et al., Extending the coverage of spectral libraries: A neighbor-based approach to predicting intensities of peptide fragmentation spectra. *PROTEOMICS* 2013, 13, 756–765.
[103] Rappsilber, J., Mann, M., What does it mean to identify a protein in proteomics? *Trends Biochem. Sci.* 2002, 27, 74–78.
[104] Nesvizhskii, A.I., Aebersold, R., Interpretation of Shotgun Proteomic Data The Protein Inference Problem. *Mol. Cell. Proteomics* 2005, 4, 1419–1440.
[105] Aebersold, R., Mann, M., Mass spectrometry-based proteomics. *Nature* 2003, 422, 198–207.
[106] Martens, L., Hermjakob, H., Proteomics data validation: why all must provide data. *Mol. Biosyst.* 2007, 3, 518–522.
[107] Daan Fierens, Learning Directed Probabilistic Logical Models from Relational Data. KU Leuven, 2008.
[108] Gupta, N., Pevzner, P.A., False Discovery Rates of Protein Identifications: A Strike against the Two-Peptide Rule. *J. Proteome Res.* 2009, 8, 4173–4181.
[109] Serang, O., Noble, W., A review of statistical methods for protein identification using tandem mass spectrometry. *Stat. Interface* 2012, 5, 3–20.
[110] Li, Y.F., Radivojac, P., Computational approaches to protein inference in shotgun proteomics. *BMC Bioinformatics* 2012, 13, S4.
[111] Nesvizhskii, A.I., A survey of computational methods and error rate estimation procedures for peptide and protein identification in shotgun proteomics. *J. Proteomics* 2010, 73, 2092–2123.
[112] Hoopmann, M.R., Moritz, R.L., Current algorithmic solutions for peptide-based proteomics data generation and identification. *Curr. Opin. Biotechnol.* 2013, 24, 31–38.
[113] Feng, J., Naiman, D.Q., Cooper, B., Probability model for assessing proteins assembled from peptide sequences inferred from tandem mass spectrometry data. *Anal. Chem.* 2007, 79, 3901–3911.
[114] Nesvizhskii, A.I., Keller, A., Kolker, E., Aebersold, R., A Statistical Model for Identifying Proteins by Tandem Mass Spectrometry. *Anal. Chem.* 2003, 75, 4646–4658.
[115] Nesvizhskii, A.I., Vitek, O., Aebersold, R., Analysis and validation of proteomic data generated by tandem mass spectrometry. *Nat. Methods* 2007, 4, 787–797.
[116] Huang, T., He, Z., A linear programming model for protein inference problem in shotgun proteomics. *Bioinforma. Oxf. Engl.* 2012, 28, 2956–2962.
[117] Huang, T., Gong, H., Yang, C., He, Z., ProteinLasso: A Lasso regression approach to protein inference problem in shotgun proteomics. *Comput. Biol. Chem.* 2013, 43, 46–54.

[118] Zhang, B., Chambers, M.C., Tabb, D.L., Proteomic parsimony through bipartite graph analysis improves accuracy and transparency. *J. Proteome Res.* 2007, 6, 3549–3557.
[119] Bandeira, N., Tsur, D., Frank, A., Pevzner, P.A., Protein identification by spectral networks analysis. *Proc. Natl. Acad. Sci. U. S. A.* 2007, 104, 6140–6145.
[120] Serang, O., Noble, W.S., Faster mass spectrometry-based protein inference: junction trees are more efficient than sampling and marginalization by enumeration. *IEEEACM Trans. Comput. Biol. Bioinforma. IEEE ACM* 2012, 9, 809–817.
[121] Spivak, M., Weston, J., Tomazela, D., MacCoss, M.J., Noble, W.S., Direct maximization of protein identifications from tandem mass spectra. *Mol. Cell. Proteomics MCP* 2012, 11, M111.012161.
[122] Park, C.Y., Klammer, A.A., Käll, L., MacCoss, M.J., Noble, W.S., Rapid and Accurate Peptide Identification from Tandem Mass Spectra. *J. Proteome Res.* 2008, 7, 3022–3027.
[123] Daelemans, Walter; Morik, Katharina (Eds.), *Machine Learning and Knowledge Discovery in Databases - European Conference, Antwerp, Belgium,*, n.d.
[124] Yang, Y., Harpale, A., Ganapathy, S., in:, Buntine W, Grobelnik M, Mladenić D, Shawe-Taylor J (Eds.), *Mach. Learn. Knowl. Discov. Databases*, Springer Berlin Heidelberg, 2009, pp. 554–569.
[125] Alves, P., Arnold, R.J., Novotny, M.V., Radivojac, P., et al., Advancement in protein inference from shotgun proteomics using peptide detectability. *Pac. Symp. Biocomput. Pac. Symp. Biocomput.* 2007, 409–420.
[126] Dost, B., Bandeira, N., Li, X., Shen, Z., et al., Accurate mass spectrometry based protein quantification via shared peptides. *J. Comput. Biol. J. Comput. Mol. Cell Biol.* 2012, 19, 337–348.
[127] Kearney, P., Butler, H., Eng, K., Hugo, P., Protein identification and Peptide expression resolver: harmonizing protein identification with protein expression data. *J. Proteome Res.* 2008, 7, 234–244.
[128] Claassen, M., Inference and Validation of Protein Identifications. *Mol. Cell. Proteomics* 2012, 11, 1097–1104.
[129] Paulovich, A.G., Billheimer, D., Ham, A.-J.L., Vega-Montoto, L., et al., Interlaboratory Study Characterizing a Yeast Performance Standard for Benchmarking LC-MS Platform Performance. *Mol. Cell. Proteomics MCP* 2010, 9, 242–254.
[130] Ivanov, A.R., Colangelo, C.M., Dufresne, C.P., Friedman, D.B., et al., Interlaboratory studies and initiatives developing standards for proteomics. *PROTEOMICS* 2013, 13, 904–909.
[131] Marx, H., Lemeer, S., Schliep, J.E., Matheron, L., et al., A large synthetic peptide and phosphopeptide reference library for mass spectrometry-based proteomics. *Nat. Biotechnol.* 2013, 31, 557–564.
[132] Orchard, S., Data Standardization and Sharing-The work of the HUPO-PSI. *Biochim. Biophys. Acta* 2013.
[133] Narayanan, A., Shi, E., Rubinstein, B.I.P., in:, *2011 Int. Jt. Conf. Neural Networks IJCNN*, 2011, pp. 1825–1834.

[134] Sing, T., Sander, O., Beerenwinkel, N., Lengauer, T., ROCR: visualizing classifier performance in R. *Bioinformatics* 2005, 21, 3940–3941.

Figure Legends

Figure 1: An illustration of how the availability of unlabeled examples can help in finding the correct "border" (boundary) between classes. A + in the figure denotes a positive example, a − a negative one, while a dot is an unlabeled instance. If one needs to draw a line that separates the positive from the negative examples, a different line would seem to be the most natural in (a) and (b). The actual boundary here lies outside of the densely populated areas of the feature set (domain). (Adapted from [15])

Figure 2: The Receiver-Operator curves of the three different predictors, with AUC-ROC the probability that the classifier will score a randomly drawn positive sample higher than a randomly drawn negative sample. Curves generated with ROCR [134].

Figure 3: Comparison of peptide detectability classifiers based on a ROC-analysis. For the iPRG data set, the area under the curve for ESPPredictor and STEPP are similar, ESPPredictor performs better at low false positive rates, while STEPP is better in the high recall regime. ESPPredictor is grayed out for the CPTAC data set because the predictor is trained on data from that same study.

Figure 4: A bipartite graph representing the three layers of information in the protein inference process. Layer I being the spectral data layer, layer II the peptide sequence annotation layer (PSM) and layer III containing protein information. The edges between the layers contain the information that can be used for protein inference; examples are given under the edges.

Figure 1:

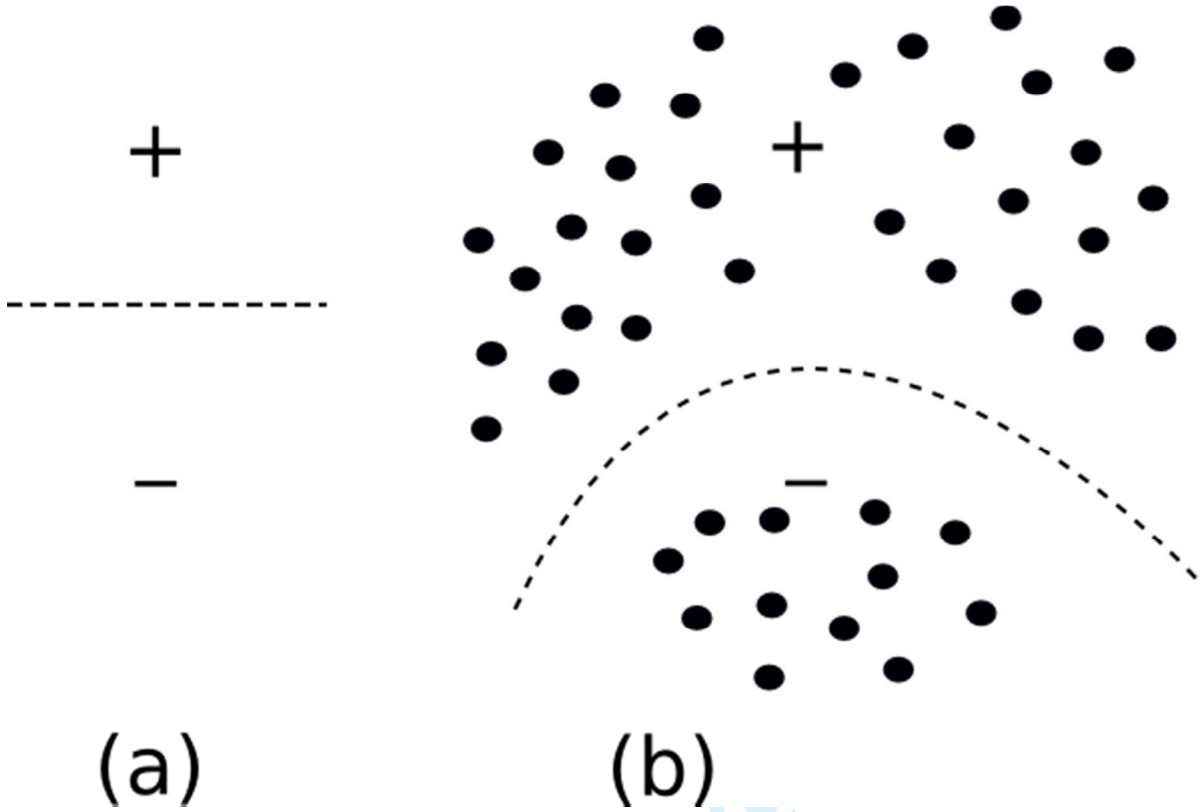


Figure 2:

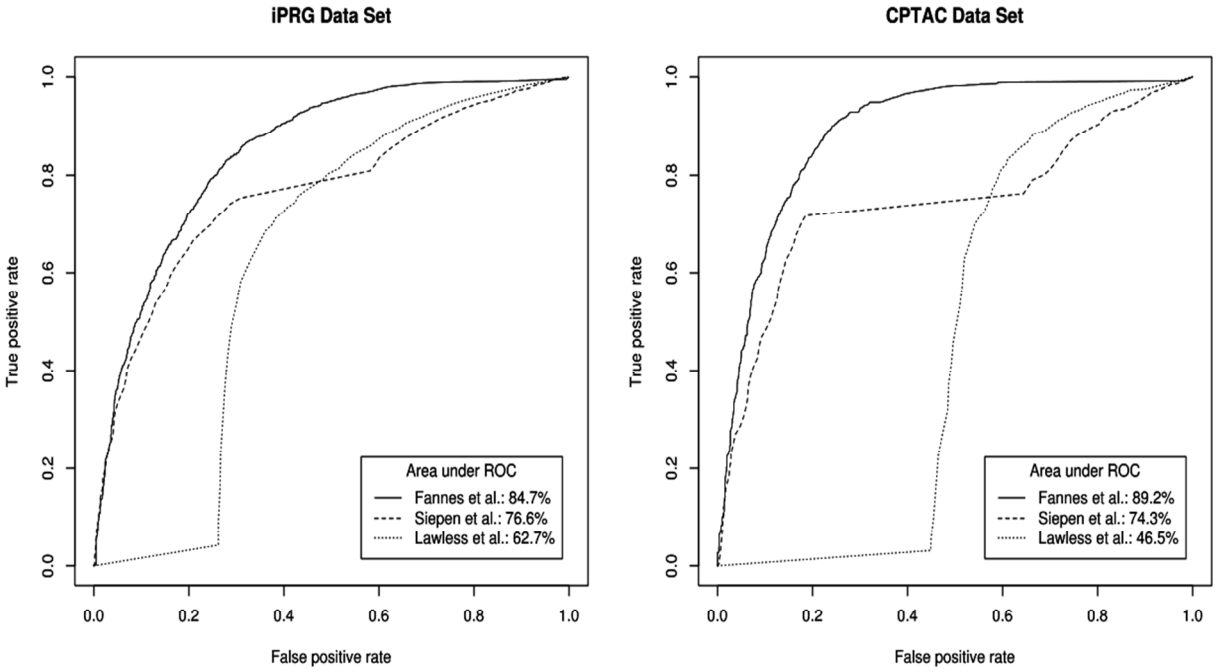


Figure 3:

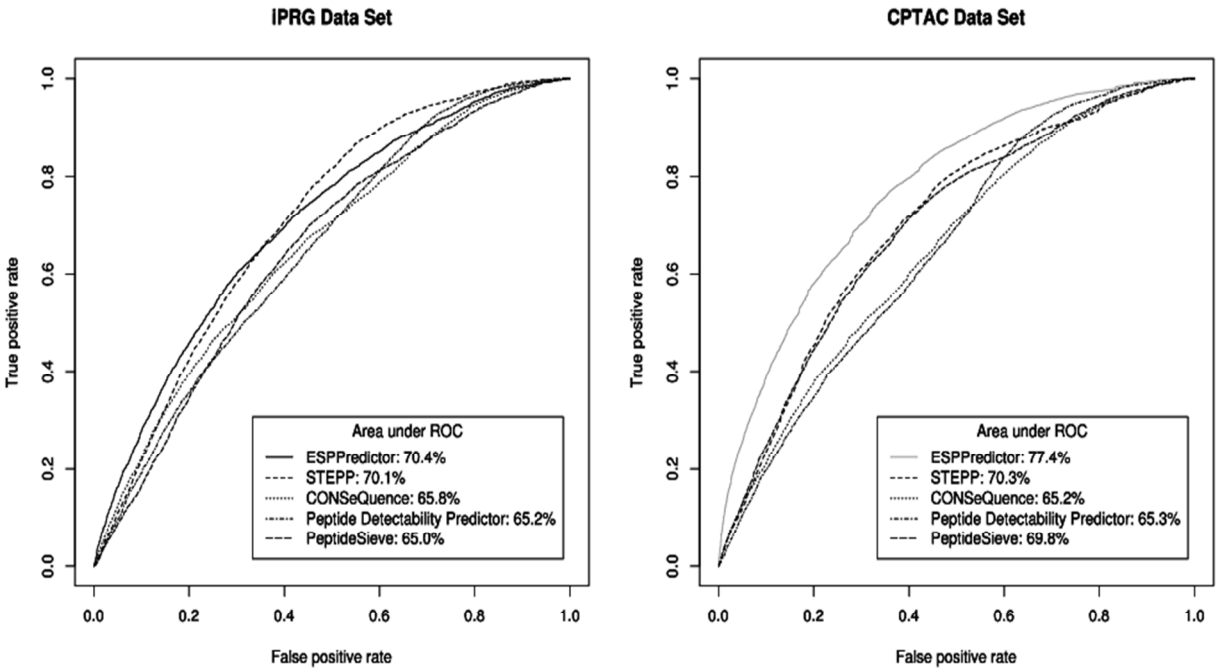
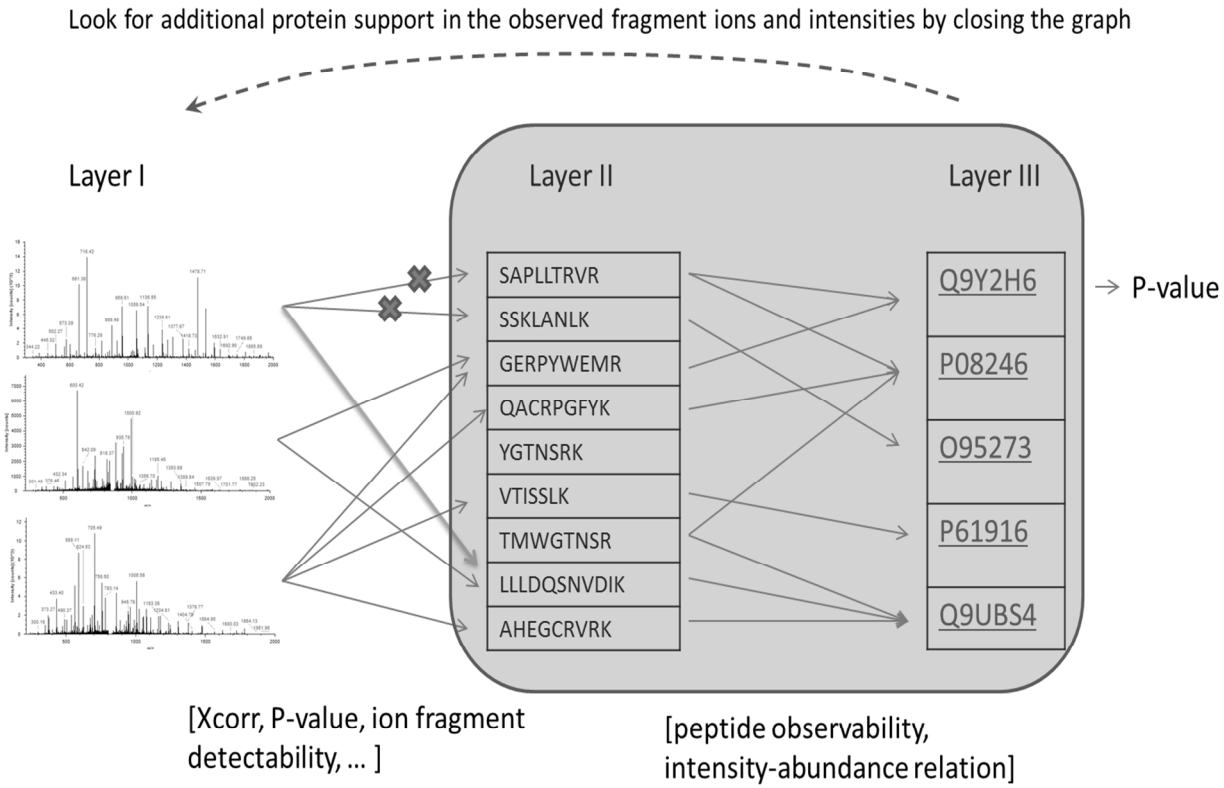


Figure 4:


Look for additional protein support in the observed fragment ions and intensities by closing the graph
Layer I
Layer II
Layer III
SAPLLTRVR
SSKLANLK
GERPYWEMR
QACRPGFYK
YGTNSRK
VTISSLK
TMWGTNSR
LLLDQSNVDIK
AHEGCRVRK
Q9Y2H6
P08246
O95273
P61916
Q9UBS4
P-value
[Xcorr, P-value, ion fragment detectability, …]
[peptide observability, intensity-abundance relation]

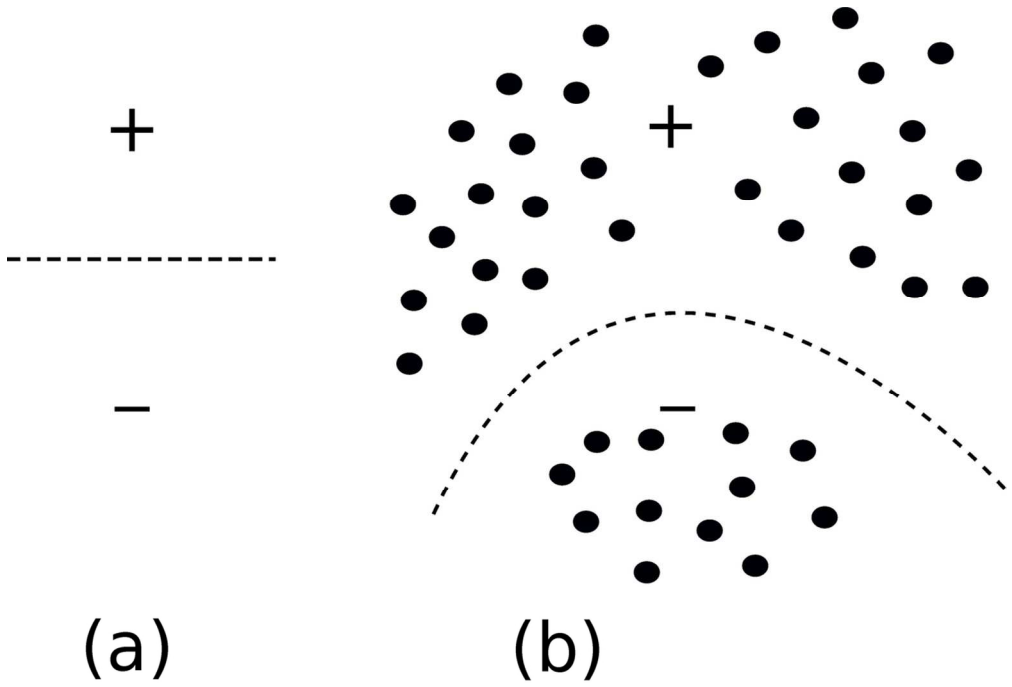


Figure 1: An illustration of how the availability of unlabeled examples can help in finding the correct "border" (boundary) between classes. A + in the figure denotes a positive example, a - a negative one, while a dot is an unlabeled instance. If one needs to draw a line that separates the positive from the negative examples, a different line would seem to be the most natural in (a) and (b). The actual boundary here lies outside of the densely populated areas of the feature set (domain). (Adapted from [15])
113x76mm (300 x 300 DPI)

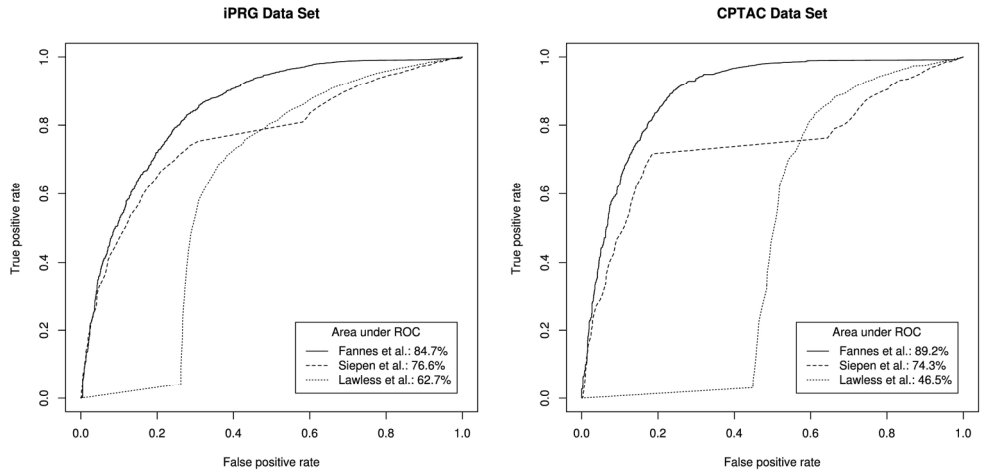


Figure 2: The Receiver-Operator curves of the three different predictors, with AUC-ROC the probability that the classifier will score a randomly drawn positive sample higher than a randomly drawn negative sample. Curves generated with ROCR [134].
177x88mm (300 x 300 DPI)

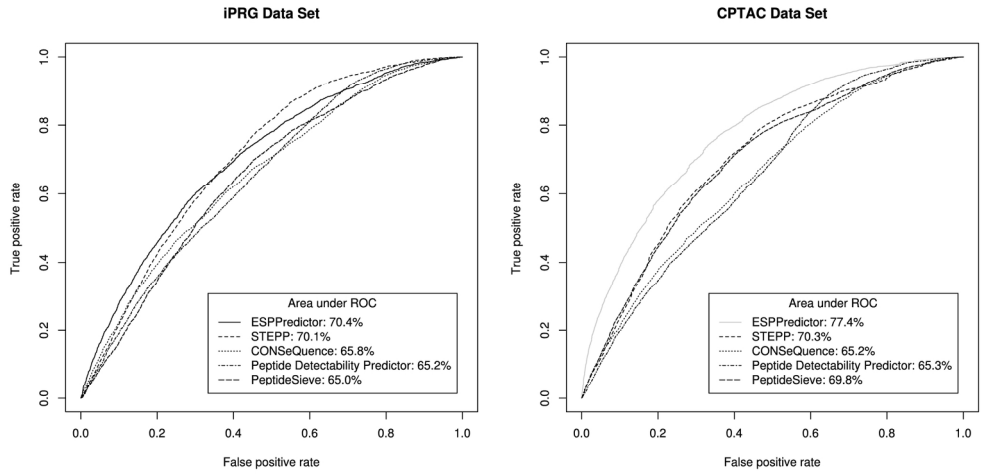


Figure 3: Comparison of peptide detectability classifiers based on a ROC-analysis. For the iPRG data set, the area under the curve for ESPPredictor and STEPP are similar, ESPPredictor performs better at low false positive rates, while STEPP is better in the high recall regime. ESPPredictor is grayed out for the CPTAC data set because the predictor is trained on data from that same study.
177x88mm (300 x 300 DPI)

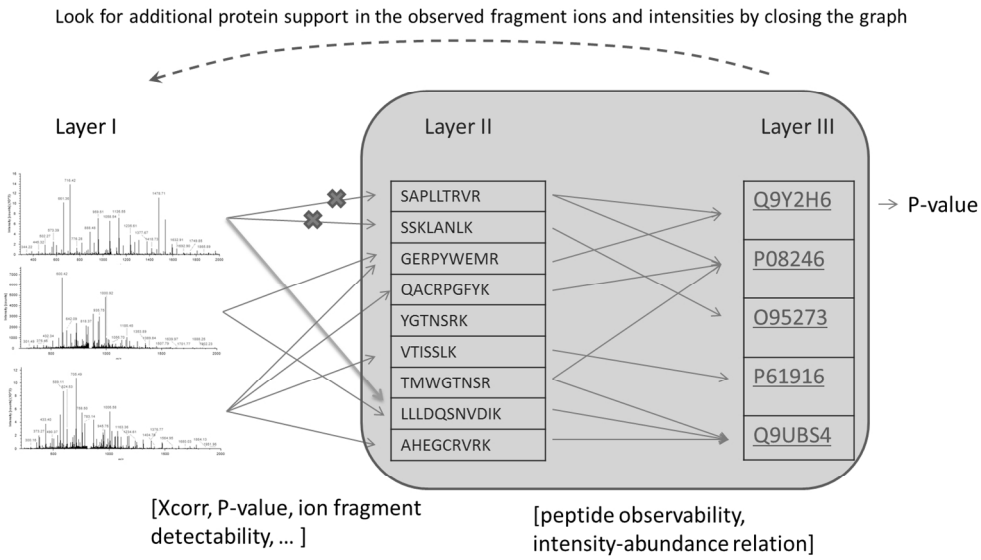


Figure 4: A bipartite graph representing the three layers of information in the protein inference process. Layer I being the spectral data layer, layer II the peptide sequence annotation layer (PSM) and layer III containing protein information. The edges between the layers contain the information that can be used for protein inference; examples are given under the edges.